

# CS 595: Assignment #10

Due on Thursday, December 11, 2014

*DR NELSON 4:20Pm*

**Victor Nwala**

## Contents

<b>Problem 1</b>	<b>3</b>
<b>Problem 2</b>	<b>6</b>
<b>Problem 3</b>	<b>10</b>
<b>Problem 4</b>	<b>11</b>

## Problem 1

1. Choose a blog or a newsfeed (or something similar as long as it has an Atom or RSS feed). It should be on a topic or topics of which you are qualified to provide classification training data. In other words, choose something that you enjoy and are knowledgeable of. Find a feed with at least 100 entries.

Create between four and eight different categories for the entries in the feed.

To answer this question I chose a politics blog which has atom xml links. I downloaded and saved the xml file with the name “politics\_search.xml”

I preprocessed in with Listing 1

Listing 1: Script to establish database and preprocess xml file

```
import docclass
import feedfilter
cl=docclass.fisherclassifier(docclass.getwords)
cl.setdb('politics_feed.db') # Only if you implemented SQLite
5 feedfilter.read('politics_search.xml',cl)
```

The read function in feedfilter.py helps me write the processed file into a file named “mainSourceData.txt” I went into the mainSourceData and manually classified my blog post to 5 categories namely: Obama–Blogs about Obama administration or him personally, General–General discussions on any other topic. Conservatives–Blogs about republicans or their ideologies. Liberals–Blogs about democrates or their ideologies. Sport–Blogs about sports or sports personalities.

Listing 2: Script to establish database and preprocess xml file

```
import feedparser
import re

5 def interestingwords(s):
    splitter = re.compile(r'\W*')
    return [s.lower() for s in splitter.split(s) if len(s) > 2 and len(s) < 20]

10 def entryfeatures(entry):
    f = {}

    # extract title
    titlewords = interestingwords(entry['title'])
15 for w in titlewords: f['Title:' + w] = 1

    # extract summary
    summarywords = interestingwords(entry['summary'])

20 # count uppercase words
    uc = 0
    for i in range(len(summarywords)):
        w = summarywords[i]
        f[w] = 1
25 if w.isupper(): uc += 1

    # get word pairs in summary as features
```

```

    if i < len(summarywords) - 1:
        twowords = ' '.join(summarywords[i:i+1])
30     f[twowords] = 1

    # keep creator and publisher as a whole
    f['Publisher:' + entry['publisher']] = 1

35     # Insert virtual keyword for uppercase words
    if float(uc) / len(summarywords) > 0.3: f['UPPERCASE'] = 1

    print f.keys()
    return f.keys()
40

def readNonInteractive(dictionaryOfTitleActualClassValues, feed, classifier):
    print '...training'
    f = feedparser.parse(feed)
    for entry in f['entries']:
45         fulltext = '%s\n%s' % (entry['title'], entry['summary'])
        #print dictionaryOfTitleActualClassValues

        titleFromXML = entry['title'].strip().lower()
        titleFromXML1 = entry['summary'].strip().lower()
50         if titleFromXML1 in dictionaryOfTitleActualClassValues:
            groundTruth = dictionaryOfTitleActualClassValues[titleFromXML1]

            print 'here'
            print titleFromXML, groundTruth
55             classifier.train(fulltext, groundTruth)

    print '...done training'

def readNonInteractiveTesting(dictionaryOfTitleActualClassValues, feed, classifier
):
60     print '...testing'

    outputFile = open('myPredictionsFile.txt', 'w')

    f = feedparser.parse(feed)
65     for entry in f['entries']:

        titleFromXML = entry['title'].strip().lower()
        titleFromXML1 = entry['summary'].strip().lower()
        fulltext = '%s\n%s' % (entry['title'], entry['summary'])
70

        if titleFromXML1 in dictionaryOfTitleActualClassValues:

            actualValue = dictionaryOfTitleActualClassValues[titleFromXML1]

75             prediction = str(classifier.classify(fulltext))
            cprobValue = classifier.getCProb()

            print actualValue, prediction, cprobValue
            outputFile.write(titleFromXML + '&' + titleFromXML1[:20] + '&' + actualValue

```

```

    + '&' + prediction + '&' + str(cprobValue) + '\\\n')
80
outputFile.close()

85
print '...done testing'

90
def read(feed, classifier):
    #out = open('mainSourceData.txt', 'w')
    f = feedparser.parse(feed)
    for entry in f['entries']:
        print
        print '----'
        print 'Title: ' + entry['title'].encode('utf-8')
        #print 'Publisher: ' + entry['publisher'].encode('utf-8')
        print
        print entry['summary'].encode('utf-8')
100
        #summary = entry['summary'].split.()
        #summary = entry['summary'].encode('ascii','ignore')
        #title = entry['title'].split.()
        #title = entry['title'].encode('ascii','ignore')
        fulltext = '%s\n%s' % (entry['title'], entry['summary'])
105
        #out.write(title + '<---->' + summary)
        #out.write('\n')
        #print 'Guess: ' + str(classifier.classify(fulltext))

        #cl = raw_input('Enter category: ')
110
        classifier.train(fulltext, cl)

        print 'Guess: ' + str(classifier.classify(fulltext))

        #cl = raw_input('Enter category: ')
115
        #classifier.train(entry, cl)
        classifier.train(fulltext, cl)
```

Note: The mainSoureData.txt is not included in my report but in github because of its size.

## Problem 2

2. Manually classify the first 50 entries, and then classify (using the fisher classifier) the remaining 50 entries. Report the `cprob()` values for the 50 titles as well. From the title or entry itself, specify the 1-, 2-, or 3-gram that you used for the string to classify. Do not repeat strings; you will have 50 unique strings.

To answer this I divided `mainSourceData.txt` in two equal halves, each containing 50 enterings, `trainingModel(dictionaryOfTitleAndClass)` is a function called to train the Model with first 50 enteries, `dictionaryOfTitleAndClass` its a dictionary of the content of the post and the respective class I assigned to it. `testingModel(dictionaryOfTitleAndClass)` on the otherhand test and make prediction based on the values supplied in the dictionary containing the last 50 enterings, the result is written to a file.

I cut the string to a length of 20, so that I could insert the result in the a table. Hence some string will not make any sense reading them.

Title	String	Actual	Prediction	cprobValue
in defense of partisan hack pundits	jonathan chait ja hr	general	obama	0.518322082932
catch of the day	to scott lemieux for	general	obama	0.341378215306
read stuff, you should	happy birthday to ja	sports	obama	0.341378215306
hey, pollsters!	three topics i'd lov	obama	obama	0.202830188679
repeal is still dead	a few bullet points	obama	obama	0.290334559941
read stuff, you should	happy birthday to ja	general	obama	0.315582819084
sunday question for liberals	same one that i used	liberals	obama	0.104802216283
sunday question for conservatives	what are you hoping	conservatives	obama	0.202830188679
my glitch is fixed!	i tweeted about this	obama	obama	0.202830188679
what mattered this week?	not much up on a tha	general	obama	0.196515281348
happy thanksgiving!	and happy hannukah,	general	general	0.196515281348
read stuff, you should	happy birthday to ja	sports	obama	0.202830188679
one more time on subsamples (ignore those polls! addendum)	last week i ja href=	obama	obama	0.202830188679
read stuff, you should	happy birthday to ja	general	general	0.735294117647
why we get majorities wrong	matt glassman ja hre	general	obama	0.315582819084
majorities	ezra klein ja href=	liberals	obama	0.341378215306
read stuff, you should	happy birthday to js	general	obama	0.735294117647
sunday question for liberals	same question, pushi	liberals	obama	0.196515281348
sunday question for conservatives	i'm going to push on	conservatives	conservatives	0.196515281348
what mattered this week?	not the only thing t	general	obama	0.315582819084
friday baseball post	i sort of think i mu	sports	obama	0.202830188679
post-nuclear etc.	just a few notes to	conservatives	obama	0.202830188679
read stuff, you should	happy birthday to ja	general	obama	0.202830188679
quick post-nuclear fizzle	the senate has gone	liberals	obama	0.202830188679
nuke (maybe) day	i'm not posting anyt	liberals	obama	0.202830188679
read stuff, you should	happy birthday to ja	general	general	0.202830188679
elsewhere/housekeeping	my tap column this w	conservatives	obama	0.202830188679
read stuff, you should	happy birthday to ja	general	obama	0.202830188679
filibuster showdown update	greg sargent is ja h	liberals	obama	0.341378215306
catch of the day	i think this properl	general	obama	0.290334559941
read stuff, you should	happy birthday to ja	general	general	0.392765801973
ignore those polls! (small sample size crosstabs edition)	josh kraushaar ja hr	liberals	obama	0.202830188679
obamacare press frenzy stuff	oy, kraushaar. oy, g	liberals	obama	0.202830188679
read stuff, you should	happy birthday to ja	sports	obama	0.571428571429
sunday question for liberals	what's the underrepo	liberals	obama	0.211617644646
sunday question for conservatives	when, if ever, do yo	conservatives	conservatives	0.235040107135
what mattered this week?	got behind on the da	general	general	0.235040107135
friday baseball post	haven't done one of	sports	obama	0.202830188679
i've seen this one before	today the house is s	conservatives	obama	0.341378215306
read stuff, you should	happy birthday to ja	general	general	0.202830188679
impeachment!	well, not the presid	conservatives	obama	0.290334559941
read stuff, you should	happy birthday to ja	general	obama	0.202830188679
gridlock/polarization	national journal is	conservatives	obama	1.0
what are republicans thinking on filibusters?	at this point in the	conservatives	obama	0.571428571429
read stuff, you should	happy birthday to ja	general	general	0.202830188679
why i expect nothing out of bauer-ginsberg	sparkd by my dismis	obama	obama	0.290334559941
plain blogger smackdown	jdiv class="tr bq" o	general	obama	0.315582819084

Listing 3: Script to train and test blog data

```
import docclass
import feedfilter

5 def trainingModel(dictionaryOfTitleAndClass):
    cl=docclass.fisherclassifier(docclass.getwords)
    cl.setdb('politics_feed.db') # Only if you implemented SQLite
    feedfilter.readNonInteractive(dictionaryOfTitleAndClass,'politics_search.xml'
        ,cl)

10 def testingModel(dictionaryOfTitleAndClass):
    cl=docclass.fisherclassifier(docclass.getwords)
    cl.setdb('politics_feed.db') # Only if you implemented SQLite
    feedfilter.readNonInteractiveTesting(dictionaryOfTitleAndClass,'
        politics_search2.xml',cl)

15 def getInput(inputFileName):
    inputFile = open(inputFileName, 'r')
    mainSourceDataLines = inputFile.readlines()

    dictionaryOfTitleAndClass = {}
    #arrayOfTitles = []

    for line in mainSourceDataLines:

        data = line.split('<----->')
        title = data[0].strip()
        classValue = data[2].strip().lower()

        title = title.lower()
        dictionaryOfTitleAndClass[title] = classValue

    arrayOfTitles.append(title)

    '''
35 arrayOfTitles.sort()
    for l in arrayOfTitles:
        print l
    '''

40 return dictionaryOfTitleAndClass

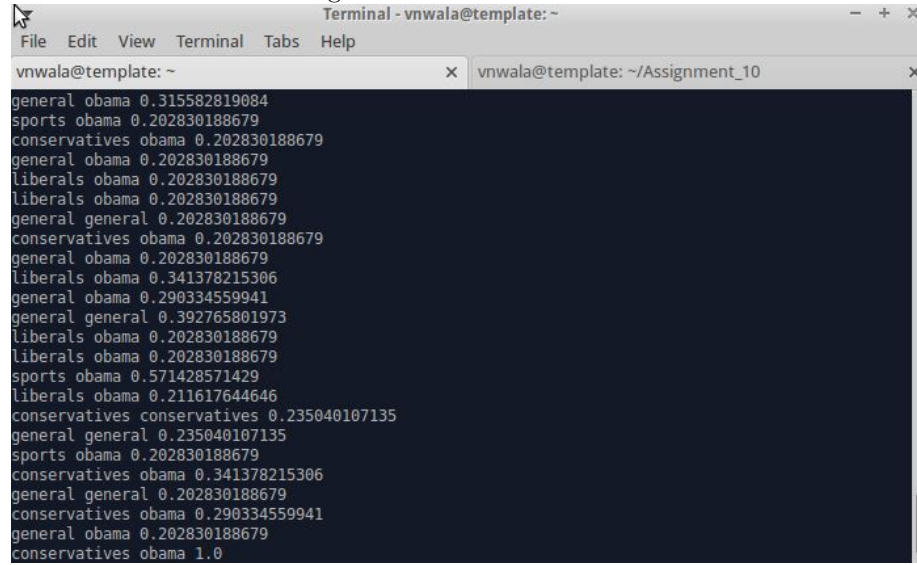
45 #dictionaryOfTitleAndClass = getInput('50FirstHalf.txt')
#print len(dictionaryOfTitleAndClass)
#trainingModel(dictionaryOfTitleAndClass)

50
```



```
dictionaryOfTitleAndClass = getInput('50SecondHalf.txt')
print len(dictionaryOfTitleAndClass)
testingModel(dictionaryOfTitleAndClass)
```

Figure 1: Prediction at work



```
vnwala@template: ~
general obama 0.315582819084
sports obama 0.202830188679
conservatives obama 0.202830188679
general obama 0.202830188679
liberals obama 0.202830188679
liberals obama 0.202830188679
general general 0.202830188679
conservatives obama 0.202830188679
general obama 0.202830188679
liberals obama 0.341378215306
general obama 0.290334559941
general general 0.392765801073
liberals obama 0.202830188679
liberals obama 0.202830188679
sports obama 0.571428571429
liberals obama 0.211617644646
conservatives conservatives 0.235040107135
general general 0.235040107135
sports obama 0.202830188679
conservatives obama 0.341378215306
general general 0.202830188679
conservatives obama 0.290334559941
general obama 0.202830188679
conservatives obama 1.0
```

### Problem 3

3. Assess the performance of your classifier in each of your categories by computing precision, recall, and F1. Note that the definitions of precisions and recall are slightly different in the context of classification.

I predicted 47 out of the 50 classes For General TP = 4, FP = 0, FN = 43, TN = 28, For Obama TP = 5, FP = 33, FN = 14, TN = 42, For Liberals TP = 0, FP = 0, FN = 47, TN = 38, For Conservatives TP = 2, FP = 0, FN = 45, TN = 38, For Sports TP = 0, FP = 0, FN = 42, TN = 47.

Class	Precision	Recall	F1
General	1	0.085	0.1569
Obama	0.13157	0.26315	0.1754
Conservatives	1	0.046	0.0816
Liberals	0	0	0
Sports	0	0	0

Also calculating the accuracy of the classification of the model with the formula  $ACC = (TP + TN) / (TP + FP + FN + TN)$ , it can be multiplied by 100 to give a percentage, I got the following results.

CLASSIFICATION	PERCENTAGE ACCURACY
General	42.66
Obama	50
Conservatives	47.05
Liberals	44.7
Sports	52.8
Average Accuracy	47.44

The average accuracy of my model is less than 50 percent.

## Problem 4

Redo questions 2 & 3, but with manually train 90 entries and then classify the remaining 10.

Title	String	Actual	Prediction	cprobValue
read stuff, you should impeachment!	happy birthday to ja well, not the presid	general conservatives	general obama	0.203045685279 0.377358490566
read stuff, you should gridlock/polarization	happy birthday to ja national journal is	general conservatives	obama obama	0.304099918545 0.189317106153
what are republicans thinking on filibusters?	at this point in the	conservatives	obama	0.203045685279
read stuff, you should	happy birthday to ja	general	general	0.203045685279
why i expect nothing out of bauer-ginsberg	sparked by my dismis	obama	obama	0.377358490566
plain blogger smackdown	jdiv class="tr_bqċo	general	obama	0.322686862035

Class	Precision	Recall	F1
General	1	0.25	0.4
Obama	0.1667	0.333	0.1538

The rest of the classes have zero Precision, Recall and F1 values

Concluding this assignment, I received some help from Alexander Nwala. Also some of the string fields in my table are not readable because it is just a part of a whole, all tuples in my table are unique.